CLAIMS:

What is claimed is:

1. A method of testing a device driver on a data processing system, said method comprising:

allocating, by an operating system, a data space for executing a device driver;

executing the device driver as an application on top of the operating system to test the device driver;

monitoring to detect whether a request made by the device driver specifies a target address within the data space; and

in response to detecting the target address for the request being made outside of the data space, trapping on the target address for the request and executing a data exception handler that emulates a target device.

2. The method according to Claim 1, further comprises:

setting up, by the operating system, the data exception handler.

3. The method according to Claim 2, wherein the setting up step further comprises:

calling the operating system to install a data exception handler facility containing the data exception handler into a vector of an interrupt table for the operating system; and

calling the data exception handler facility to register data that is used to determine when the data exception is to be taken wherein the data include each of various target addresses at which the data exception is to be taken, lengths of the target addresses, and user callback routines for calling back to the application into a data exception database table.

4. The method according to Claim 3, further comprises:

passing, by the data exception handler facility, the data space to the vector of the interrupt table.

5. The method according to Claim 4, wherein the data exception handler is a memory mapped input/output (IO) exception handler comprising a parser with a disassembler that is used for disassembling and identifying an assembler instruction of the request.

6. The method according to Claim 5, wherein the assembler instruction is identified as a load command.

7. The method according to Claim 5, wherein the assembler instruction is identified as a store command.

8. The method according to Claim 3, wherein the data exception handler further comprises:

saving, into a memory stack, data for the request which includes at least the target address for the request, content data relating to the request, and a data address range for the content data;

determining whether the target address that is saved into the memory stack is within the database exception database table;

immediately terminating the method of testing if the target address is not within the database exception database table;

disassembling the data for the request into disassembled information and passing the disassembled information into a respective one of the user callback routines stored in the data exception database table wherein the respective one of the user callback routines that is used is based on the request and is for emulating a behavior of the target device;

emulating, by the respective one of the user callback routines, the behavior of the target device;

setting a next instruction address to an address in the memory stack that is after the currently saved target address; and

unwinding the memory stack by the data exception

handler to return control back to the application.

9. The method according to Claim 8, further comprising:

copying, as necessary, the content data by the respective one of the user callback routines from another data space for the respective one of the user callback routines based on the request.

10. The method according to Claim 1, wherein the data exception handler is a software emulator of the target device and further comprises:

using a hardware emulator to test the application for the device driver if the hardware emulator is coupled to the data processing system; and

using the software emulator to test the application for the device driver if the hardware emulator is not coupled to the data processing system.

11. The method according to Claim 1, further comprises:

determining whether the application for the device driver has finished executing;

continuing with the executing of the application if the application has not finished executing; and

terminating the method of testing when the application has finished executing.

12. A data processing system for testing a device driver, said data processing system comprising:

a processor and a memory system, wherein:

said processor executes an operating system that allocates a data space for executing applications and executes a device driver as an application on top of the operating system to test the device driver; and

said processor and said memory system, responsive to detecting a request by the device driver specifying a target address outside of the data space, trap on the target address and execute a data exception handler that emulates a target device of the device driver.

13. The data processing system according to Claim 12, wherein said processor and said memory system call a data exception handler facility to register data that is used to determine when a data exception is to be taken, wherein the data include each of various target addresses at which the data exception is to be taken, lengths of the target addresses, and user callback routines for calling back to the application into a data exception database table.

14. The system according to Claim 13, wherein said processor and said memory system call the operating system to install the data exception handler facility containing the data exception handler into a vector of an interrupt table for the operating system.

15. The system according to Claim 14, wherein the data exception handler facility passes the data space to the vector of the interrupt table.

16. The system according to Claim 15, wherein the data exception handler is a memory mapped input/output (IO) exception handler that is stored in the memory system and comprises a parser with a disassembler that is used for disassembling and identifying an assembler instruction of the request.

17. The system according to Claim 16, wherein the assembler instruction is a load command.

18. The system according to Claim 16, wherein the assembler instruction is a store command.

19. The system according to Claim 14, wherein the data exception handler determines whether the target address

is within the data exception database table and, responsive to said determination, immediately terminates testing of the device driver if the target address is not within the data exception database table, and, emulates, by the respective one of the user callback routines, the behavior of the target device if the target address is within the data exception database table.

20. The system according to Claim 19, wherein the data exception handler further:

copies, as necessary, the content data by the respective one of the user callback routines from another data space for the respective one of the user callback routines based on the request.

21. The system according to Claim 12, wherein the data exception handler is a software emulator of the target device and wherein:

a hardware emulator is used to test the application for the device driver if the hardware emulator is coupled to the data processing system; and

the software emulator is used to test the application for the device driver if the hardware emulator is not coupled to the data processing system.

22. A program product for testing a device driver on a data processing system comprising:

instruction means for allocating, by an operating system, a data space for executing a device driver;

instruction means for executing the device driver as an application on top of the operating system to test the device driver;

instruction means for monitoring to detect whether a request made by the device driver specifies a target address within the data space;

in response to detecting the target address for the request being made outside of the data space, instruction means for trapping on the target address for the request and executing a data exception handler that emulates a target device; and

computer usable media bearing said instruction means.

23. The program product according to Claim 22, further comprises:

instruction means for setting up, by the operating system, the data exception handler.

24. The program product according to Claim 23, wherein the instruction means for setting up further comprises:

instruction means for calling the operating system to install a data exception handler facility containing the data exception handler into a vector of an interrupt table for the operating system; and

instruction means for calling the data exception handler facility to register data that is used to determine when the data exception is to be taken wherein the data include each of various target addresses at which the data exception is to be taken, lengths of the target addresses, and user callback routines for calling back to the application into a data exception database table.

25. The program product according to Claim 24, further comprises:

instruction means for passing, by the data exception handler facility, the data space to the vector of the interrupt table.

26. The program product according to Claim 22, wherein the data exception handler further comprises:

instruction means for saving, into a memory stack,

data for the request which includes at least the target address for the request, content data relating to the request, and a data address range for the content data;

instruction means for determining whether the target address that is saved into the memory stack is within the database exception database table;

instruction means for immediately terminating the testing of the device driver if the target address is not within the database exception database table;

instruction means for disassembling the data for the request into disassembled information and passing the disassembled information into a respective one of the user callback routines stored in the data exception database table wherein the respective one of the user callback routines that is used is based on the request and is for emulating a behavior of the target device;

instruction means for emulating, by the respective one of the user callback routines, the behavior of the target device;

instruction means for setting a next instruction address to an address in the memory stack that is after the currently saved target address; and

instruction means for unwinding the memory stack by the data exception handler to return control back to the application.

27. The program product according to Claim 26, further comprising:

instruction means for copying, as necessary, the content data by the respective one of the user callback routines from another data space for the respective one of the user callback routines based on the request.

28. The program product according to Claim 22, wherein the data exception handler is a software emulator of the target device and further comprises:

instruction means for using a hardware emulator to test the application for the device driver if the hardware emulator is coupled to the data processing system; and

instruction means for using the software emulator to test the application for the device driver if the hardware emulator is not coupled to the data processing system.

29. The program product according to Claim 22, further comprises:

instruction means for determining whether the application for the device driver has finished executing;

instruction means for continuing with the executing of the application if the application has not finished executing; and

instruction means for terminating the testing of the device driver when the application has finished executing.